1. Consider our standard model for the dynamics of a single mRNA species with copy number n, with transitions

$$\begin{aligned} n &\rightarrow n+1 \\ n &\rightarrow n-1 \end{aligned}$$

and corresponding rates β and αn. Use parameter values $\beta = 1/\text{min}$ and $\alpha = 0.1/\text{min}$.

(a) What is average lifetime of a transcript in minutes? For stochastic dynamics at equilibrium, what are the theoretical values for the mean and variance of n?
Consider a single transcript which can only degrade with rate α. The time until degradation is distributed exponentially with parameter α (ie $T \sim exp(\alpha)$). The exponential distribution is given by:

$$f(t) = \begin{cases} \alpha e^{\alpha t} & t \geq 0 \\ 0 & t < 0 \end{cases}$$

The average lifetime of a transcript is the expected value of the time until a single transcript degrades. Therefore:

$$t_{AVG} = \mathbb{E}[T] = \int_{-\infty}^{\infty} t f(t) dt = \int_{0}^{\infty} t \alpha e^{\alpha t} = \frac{1}{\alpha}$$

Therefore the average lifetime of a transcript is:

$$\frac{1}{\alpha} = 10 \text{ mins}$$

At equilibrium, assume that P_n is the probability that there are n copies of transcripts. Then $\sum_{i=1}^{+\infty} P_i = 1$. And

$$\begin{aligned} P_0\beta &= P_1\alpha \\ P_1\beta &= P_2(2\alpha) \\ &\cdots \\ P_{n-1}\beta &= P_n(n\alpha) \\ &\cdots \end{aligned}$$

We can solve the probability P_n

$$P_n = \frac{1}{n!}\left(\frac{\beta}{\alpha}\right)^n e^{-\frac{\beta}{\alpha}}$$

So the copy number n have a Poisson distribution, and The mean of n is

$$\langle n \rangle = \frac{\beta}{\alpha} = 10$$

And the variance of n is

$$Var(n) = \frac{\beta}{\alpha} = 10$$

(b) Using the variable X to denote a continuous approximation to the particle number n, what is the ODE corresponding to the stochastic dynamics? What is the steady-state value of X, defined as $\langle X \rangle$?
The ODE function is

$$\frac{dX}{dt} = \beta - \alpha X$$
$$\frac{dX}{dt} = 1 - \frac{X}{10}$$

At steady-state, $dX/dt = 0$, so the value of X would be

$$\langle X \rangle = \frac{\beta}{\alpha} = 10$$

(c) Write a Gillespie simulation for this system and provide your source code. A code framework will be available as a starting point. Perform a stochastic simulation for the dynamics of n using your simulation software. Have the program output the value of n at regular intervals of 1 minute. This takes some care because the random transitions will not necessarily occur at these fixed intervals. You do not have to provide the trajectory, just the source code.
The code is attached to the end of this key

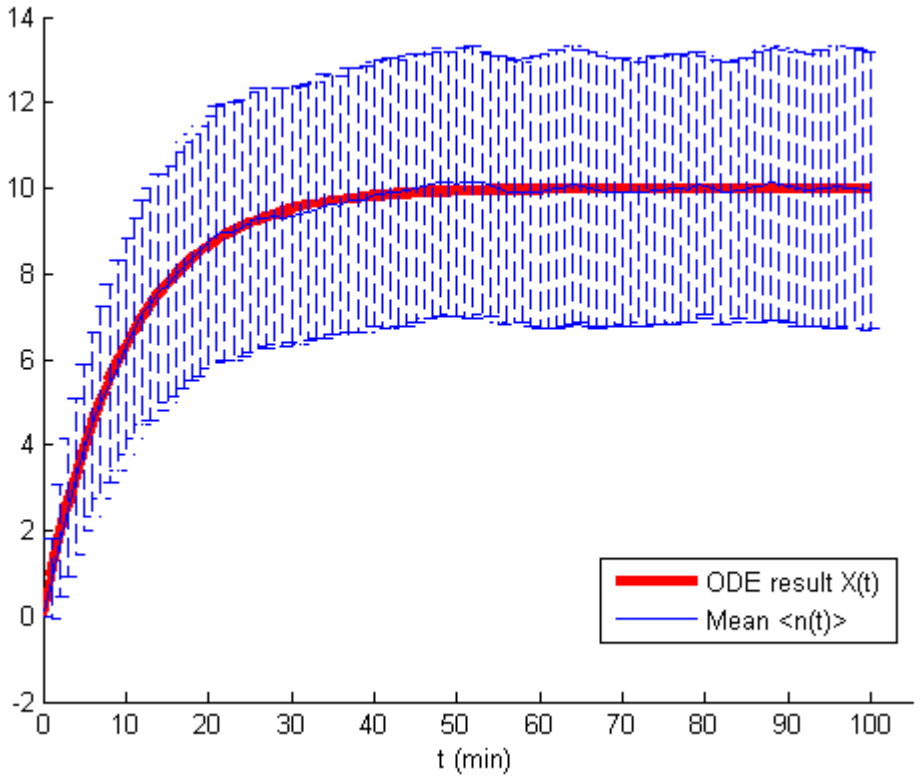
(d) Start the simulation at n equal to the integer closest to $\langle X \rangle$. Calculate and report the mean μ and variance σ^2 of n over 10,000 minutes. Compare these to the theoretical values.

2. Now consider the approach to equilibrium. The dynamical systems are the same, but the initial conditions are $X = 0$ (ODE) and $n = 0$ (stochastic).

(a) What is the solution for $X(t)$?
Based on the ODE function we got in question $1(b)$ and the initial condition $X = 0$, we can solve the response $X(t)$ as

$$\begin{aligned} X(t) &= \frac{\beta}{\alpha}\left(1 - e^{-\alpha t}\right) \\ &= 10\left(1 - e^{-t/10}\right) \end{aligned}$$

(b) Run 1000 simulations (trajectories) of the first $T = 100$ minutes for the stochastic system. Calculate the mean $\langle n(t)\rangle$ and the variance $\mathrm{Var}[n(t)]$ at 1 minute intervals. Note: The mean $\langle n(t)\rangle$ here depends on time; it is computed by averaging over the 1000 trajectories at each time point. Similarly, the variance $\langle n(t)^2\rangle - \langle n(t)\rangle^2$ is evaluated at 1 minute intervals over the 1000 simulations, computed separately for each time t. It is not the variance over the 100 minute trajectory. You should find that $\langle n(0)\rangle = \mathrm{Var}[n(0)] = 0$, $\langle n(T)\rangle = \mu$, and $\mathrm{Var}[n(T)] = \sigma^2$, where μ and σ^2 are from the previous equilibrium simulation. Plot the mean (solid line) and $\pm$ one standard deviation error bars (dashed line, standard deviation computed simply as the square root of the variance) for the 1000 trajectories, together with the ODE results for $X(t)$ (thick line).



The code is attached to the end the file.

(c) At what time t is the variance maximum? Can you suggest (or derive) an analytical expression for the time when the variance is maximum?

By looking at the plotting in question 2(b) and its trend, we can see that the variance reaches the maximum when response X(t) is approaching the equilibrium, so the expression for the time is $t \to \infty$.

3. Finally, consider a model with bursts of transcription with burst size 5. The stochastic transitions are

$$\begin{aligned} n &\to n+5 \\ n &\to n-1 \end{aligned}$$

with corresponding rates $\beta/5$ and α.

(a) What is the corresponding ODE model? Does the ODE model depend on the burst size?

$$\frac{dX}{dt} = \frac{\beta}{5} \cdot 5 - \alpha X$$

For the ODE solution there is no dependence on the burst size.

(b) Start a stochastic simulation with n equal to the closest integer to the steady-state value for the ODE model, and generate a 10,000 minute trajectory, recording the state (the value of n) at 1-minute intervals. Calculate and report the mean μ and variance σ^2. Approximately:

$$\begin{aligned} \hat{\mu} &= 10.17 \\ \hat{\sigma}^2 &= 30.44 \end{aligned}$$

(c) How do the mean and variance compare to the results for the stochastic system with burst size of 1? Do larger bursts correspond to smaller, equal, or larger mean and variance? Can you provide a conjecture for the mathematical expression relating mean and variance for bursts of size b generated at rate β/b, together with the rates β and α? Running the simulation for different values of b you can compute the table:

b	σ^2
1	≈ 10
5	≈ 30
7	≈ 40
9	≈ 50
10	≈ 55

By constructing the above table you will notice a pattern: the variance seems to be linearly related to b. You might postulate that the relationship will have the form:

$\sigma^2 = s \cdot b + v$ where s is the slope and v is the intercept. Now consider the case b = 1. In that case you know that the analytical solution is: $\sigma^2 = \frac{\beta}{\alpha}$. Using these facts a reasonable conjecture for the variance in terms of parameters is:

$$\sigma^2 = \frac{b+1}{2}\frac{\beta}{\alpha}$$

Matlab Source Code

Prob1.m

```
function Prob1()
global a b
a = .1; % alpha
b = 1; % beta

n_0 = 10; % Initial count
nt = 1; % Number of trajectories
t_max = 10; %Minutes for simulation (Approx).
G = cell(nt,2);
figure(1);
hold on
for i = 1:nt
    [T, N] = get_trajectory(n_0, t_max);
    G{i,1} = T;
    G{i,2} = N;
    stairs(G{i,1},G{i,2})
end
%hold off

%figure(2);
[T_int, N_int] = const_intervals(T, N, t_max, 1);
plot(T_int,N_int,'.');
end

function [T, N] = get_trajectory(n_0, t_max)
%T - transition times
%N - counts at each transition time
global a b
t = 0;
j = 1;
T(1) = 0;
N(1) = n_0;
while t < t_max
    j = j+1;
    dt = (-1/(a*N(j-1) + b)) * log(rand(1));
    t = t + dt;
    T(j) = t;
```

```
    if rand(1) < b/(b+a*N(j-1))
        N(j) = N(j-1) + 1;
    else
        N(j) = N(j-1) - 1;
    end
end
end

function [T_int, N_int] = const_intervals(T, N, t_max, dt_int)

% T_int - Times at even intervals.
% N_int - Count at times corresponding to T_int.
% T - Vector of transition times.
% N - Count of mRNA corresponding to times given in T.
% t_max - maximum of amount time
% dt_int - size of time interval
T_int = 0:dt_int:t_max;
N_int = size(T_int);
N_int(1) = N(1);
j = 1;
for i = 2:length(T_int)
    while T(j) < T_int(i)
        j = j+1;
    end
    N_int(i) = N(j-1);
end
end
```

Prob2.m

```
function Prob2
global a b
a = .1; % alpha
b = 1; % beta

n_0 = 0; % Initial count
nt = 1000; % Number of trajectories
t_max = 100; %Minutes for simulation (Approx).
G = cell(nt,2);
K = zeros(nt,t_max+1);
Var_int=zeros(1,t_max+1);

for i = 1:nt
    [T, N] = get_trajectory(n_0, t_max);
    G{i,1} = T;
    G{i,2} = N;
    [T_int, N_int] = const_intervals(T, N, t_max, 1);
    K(i,:)=N_int;
end
Mean_int=mean(K);

for j=1:t_max+1
    Var_int(j)=mean(K(:,j).^2)-mean(K(:,j))^2;
end

X=(b/a).*(1.-exp(-a.*T_int));
figure
hold on
h1=plot(T_int,X);
set(h1,'LineWidth',4,'Color','red');
h2=errorbar(T_int,Mean_int,sqrt(Var_int));
h2Children = get(h2, 'children');
set(h2Children(2), 'linestyle','--')
xlim([0,105])
legend('ODE result X(t)','Mean <n(t)>')
xlabel('t (min)')
[M,I]=max(Var_int)
end
```

```
function [T, N] = get_trajectory(n_0, t_max)
%T - transition times
%N - counts at each transition time
global a b
t = 0;
j = 1;
T(1) = 0;
N(1) = n_0;
while t < t_max
    j = j+1;
    dt = (-1/(a*N(j-1) + b)) * log(rand(1));
    t = t + dt;
    T(j) = t;
    if rand(1) < b/(b+a*N(j-1))
        N(j) = N(j-1) + 1;
    else
        N(j) = N(j-1) - 1;
    end
end
end

function [T_int, N_int] = const_intervals(T, N, t_max, dt_int)

% T_int - Times at even intervals.
% N_int - Count at times corresponding to T_int.
% T - Vector of transition times.
% N - Count of mRNA corresponding to times given in T.
% t_max - maximum of amount time
% dt_int - size of time interval
T_int = 0:dt_int:t_max;
N_int = size(T_int);
N_int(1) = N(1);
j = 1;
for i = 2:length(T_int)
    while T(j) < T_int(i)
        j = j+1;
    end
    N_int(i) = N(j-1);
end
end
```

Prob3.m

```
function Prob3()
global a rf burst
a = .1; % alpha
burst = 5;
b = 1; % beta
rf = b/burst; %The forward rate of the reaction. This replaces beta below.

n_0 = 1; % Initial count
nt = 1; % Number of trajectories
t_max = 10000; %Minutes for simulation (Approx).

G = cell(nt,2);
figure(1);
hold on
for i = 1:nt
    [T, N] = get_trajectory(n_0, t_max);
    G{i,1} = T;
    G{i,2} = N;
    stairs(G{i,1},G{i,2})
end
[T_int, N_int] = const_intervals(T, N, t_max, 1);
mean(N_int)
var(N_int)
plot(T_int,N_int,'.');
hold off
end

function [T, N] = get_trajectory(n_0, t_max)
% GET_TRAJECTORY - Computes a single Gillespie trajectory.
% Outputs:
% T - transition times
% N - counts at each transition time
% Inputs:
% n_0 - initial number of molecules.
% t_max - length of time of the simulation.

global a rf burst
t = 0;
j = 1;
```

```
T(1) = 0;
N(1) = n_0;
while t < t_max
    j = j+1;
    dt = (-1/(a*N(j-1) + rf)) * log(rand(1));
    t = t + dt;
    T(j) = t;
    if rand(1) < rf/(rf+a*N(j-1))
        N(j) = N(j-1) + burst;
    else
        N(j) = N(j-1) - 1;
    end
end
end

function [T_int, N_int] = const_intervals(T, N, t_max, dt_int)
% CONST_INTERVALS - Takes a trajectory as input. Then computes the value of
% n at even intervals.

% Outputs:
% T_int - Times at even intervals.
% N_int - Count at times corresponding to T_int.
% Inputs:
% T - Vector of transition times.
% N - Count of mRNA corresponding to times given in T. A single trajectory.
% t_max - maximum of amount time
% dt_int - size of time interval

T_int = 0:dt_int:t_max;
N_int = size(T_int);
N_int(1) = N(1);
j = 1;
for i = 2:length(T_int)
    while T(j) < T_int(i)
        j = j+1;
    end
    N_int(i) = N(j-1);
end
end
```